Marked-Up Version of the Amended Specification

Page 1, line 18 through page 2, line 4:

SUB C1 Over the past several years access servers (also referred to as communication servers or terminal servers) have been used to receive data communications and to route data from remote locations onto networks such as the Internet. For example, Internet ~~Server~~ Service Providers (ISPs) typically use access servers to administer data communications from ISP subscribers. In such implementations, the ISP typically configures one or more access servers in connection with modems, which are connected to phone lines. ISP customers, who maintain their own computers with modems, establish a connection to the ISP by placing an ordinary telephone call from their home modem to the ISP modem. The ISP modem provides data to the access server, which typically authenticates the user and facilitates a connection from the users PC to the Internet across the modem-to-modem telephone connection. Such systems may also be used to access, for example, corporate intranets or the like.

A1

Page 2, lines 5-9:

SUB C2 Recent advances in access server hardware has allowed the access server to support not only modem connections, but also fax, video conference, voice, multimedia, Asynchronous Transfer Mode (ATM), frame relay, and other types of connections. Such systems frequently include data communications processors such as the Any Port Products available from Conexant Systems of Newport Beach, California.

A2

Page 3, lines 9-17:

SUB C3 Another option is to develop a device driver that is unique for the particular access server or access server components utilized. Again, however, this approach provides a proprietary solution that is unique to the particular application or hardware device included. Such systems

A3

are slow to incorporate new functionality in operating systems or hardware, and moreover, they do not facilitate direct addressability from the external host. Direct accessibility is particularly desirable in environments with distributed gateways for routing voice and data communications. Hence, with the "gateway decomposition" schemes currently pursued many systems providers typically require proprietary device drivers or APIs for interacting with particular hardware.

Page 8, lines 11-19:

With reference to Figure 1, an access server system 100 suitable for ISDN or modem communications suitably includes a communications interface 104, access hardware 120, and a computer host 140. The communications interface 104 is any interface device that suitably receives call information from a network 102 such as the public switched telephone network (PSTN) and provides the call information via a bus 106 to an access device. The access device may be implemented, for example, as access hardware 120, as a modem (not shown), or as an ISDN modem 114. In various embodiments of the invention, communications interface 104 is a T1 interface, a modem, or another form of communications service unit (CSU).

Page 9, line 21 through page 10, line 10:

Communications may be suitably controlled by a an application program 134 running on host 140. Exemplary computer applications include Remote Access Server (RAS) available from the Microsoft Corporation of Redmond, Washington or the pppd ("PPP daemon") included with various versions of the LINUX or UNIX operating systems. In such embodiments, application program 134 suitably receives requests for network services, responds to such requests by establishing connections, and assigns a network address (such as an IP address) to the connection established. Some embodiments of application program 134 further provide authentication or other security services. Of course, it will be understood that application

program 134 may be any suitable communications server or controller program, and that the particular functionality and implementation of application program 134 will vary from embodiment to embodiment.

Page 11, lines 6-16:

Applications such as application program 134 running on a host frequently interact with device drivers through an intermediate application programming interface (API) 136 that provides a particular type of functionality depending upon the particular device driver. ~~Interface~~ API 136, for example, may be a telephony application interface (TAPI) such as TAPI 2.1 included with Microsoft Windows NT version 4.0. Telephony API 136 suitably implements many common functions required by application program 134 related to telephony such as modem controls, ISDN controls, and the like. To control a modem via application 134, for example, a programmer suitably includes computer instructions in application 134 that interact via interface 146 with telephony API 136 to control device driver 122, which in turn interacts with access hardware 120 and/or ISDN device 114.

Page 11, line 17 through page 12, line 3:

Similarly, application program 134 interacts with a network interface 124 via network API 130 such as, for example, the NDIS 4.2 API provided with the Microsoft Windows NT operating system. Various embodiments of the network API 130 suitably include a protocol stack 132 that implements a particular suite of communications protocols such as TCP/IP, IPX, or the like. Applications 134 do not typically interact directly with network interface 124, ~~then,~~ but rather include calls to network API 130 via interface 144. Network ~~interface~~ API 130 then suitably formats data packets in accord with a protocol suite 132 and provides the packets to network interface 124 for transmission on network 126.

Page 12, lines 4-16:

SUB C8 As described above, application program 134 interacts with access hardware 120 and network interface 124 via device drivers 122 and 124 and via APIs 136 and 130, respectively. An illustrative example will show how the various elements of an exemplary embodiment work together to implement an access server system 100. When a call is received at telephony interface 104, for example, PCM data is suitably transmitted via bus 106 to the access hardware A8 120. A processor in access hardware 120 recognizes the incoming call as a modem call, for example, and modem module 116 places a request to an administering application (which may be application program 134) to create a modem session with the access server system 100. Modem module 116 sends a request for a new connection via device driver 122, which forwards the request to application program 134 through telephony API 136. The application program 134 receives the request and administers the new session with the access hardware 120, again by sending commands and gathering data via telephony API 136.

Page 13, lines 11-15:

SUB C9 As noted above, exemplary embodiments of telephony API 136 contain adequate controls for interacting with modems and modem modules such as module 116 on access hardware 120. A9 As such, modem connections are relatively easy to implement in access server system 100. Telephony API 136 does not, however, typically contain controls for accessing voice modules in access hardware 120.

Page 14, lines 1-11:

SUB C10 Access hardware 220 communicates with host computer 140 through device driver 222, A10 which is similar to device driver 122 described above but includes added support for voice module 202. Device driver ~~Driver~~ 222 provides an interface between access hardware 220 and

host 140 through, for example, conventional interface techniques. Although Figure 2 shows device driver 222 as part of host 140, it should be noted that device driver 222 functionality may be suitably implemented as software on host 140 or in hardware, software or firmware of access hardware 220. Alternatively, device driver 222 can be implemented as any combination of hardware, software and firmware on access hardware 220 and/or host 140. Various embodiments of the invention implement device driver 222 as a Windows NT miniport or as a LINUX or UNIX device driver, although of course any suitable hardware or software interface could be used.

Page 14, line 12 through page 15, line 2:

In various embodiments of the invention, voice module 202 is represented to host 140 as a modem connection. Various embodiments implement the modem-like connection differently, but exemplary techniques include formulating a modem connection in device driver 222 such that operating system 138 "thinks" that data sessions with voice module 202 are modem connections instead of voice connections. To this end, device driver 222 suitably presents voice module 202 to host 140 as a modem port. In various embodiments of the invention, the modem connection is established by passing an electronic message to telephony API 136 with modem parameters instead of conventional voice parameters. These parameters are dictated by the particular telephony API, and vary from implementation to implementation. When a connection is initiated in this manner, telephony API 136 suitably creates a connection with device driver 222 that emulates a virtual modem connection 230 between voice module 202 and telephony API 136.

Page 15, line 12 through page 16, line 11:

With continued reference to Figure 2, ~~voice~~ access server system 200 suitably administers voice calls by providing a convenient interface between application program 134 and voice module 202 such that voice data is effectively transported to network 126. When voice service is initiated (for example at startup, or as directed by application program 134, or in response to a voice call received from network 102), module 202 establishes a virtual modem connection 230 through driver 222 and operating system 138 to application program 134. Because the virtual connection 230 acts as a modem connection, application program 134 communicates to device driver 222 through conventional telephony API calls. These calls are suitably addressed to device driver 222, which converts instructions and data to a format that is understood by module 202 as necessary. In this manner, application program 134 suitably interacts via telephony API 136 and network API 130 to create virtual connection 148 between access hardware 220 and network 126 using, for example, techniques similar to those employed to create the virtual modem connections described above. In various embodiments, application program 134 assigns a network address (such as an IP address) to the connection such that virtual connection 148 (and thus the port on access server 220 corresponding to voice module 202) is addressable by entities on network 126. With the virtual connection established, voice communications are suitably routed between server 220 and network 126. Thus, a standard computer application 134 such as RAS or pppd can be used in conjunction with a standard telephony API 136 to implement an access server system 200 that provides voice functionality.